Graph II

The **order-zero graph**, K_0, is the unique graph having no vertices (hence its order is zero). It follows that K_0 also has no edges. Some authors exclude K_0 from consideration as a graph (either by definition, or more simply as a matter of convenience). Whether including K_0 as a valid graph is useful depends on context. On the positive side, K_0 follows naturally from the usual set-theoretic definitions of a graph (it is the ordered pair (*V*, *E*) for which the vertex and edge sets, *V* and *E*, are both empty), in proofs it serves as a natural base case for mathematical induction, and similarly, in recursively defined data structures K_0 is useful for defining the base case for recursion (by treating the **null tree** as the child of missing edges in any non-null binary tree, every non-null binary tree has *exactly* two children). On the negative side, including K_0 as a graph requires that many well-defined formulas for graph properties include exceptions for it (for example, either "counting all strongly connected components of a graph" becomes "counting all *non-null* strongly connected components of a graph", or the definition of connected graphs has to be modified not to include K_0). To avoid the need for such exceptions, it is often assumed in literature that the term *graph* implies "graph with at least one vertex" unless context suggests otherwise.[1][2]

Subgraphs and minors

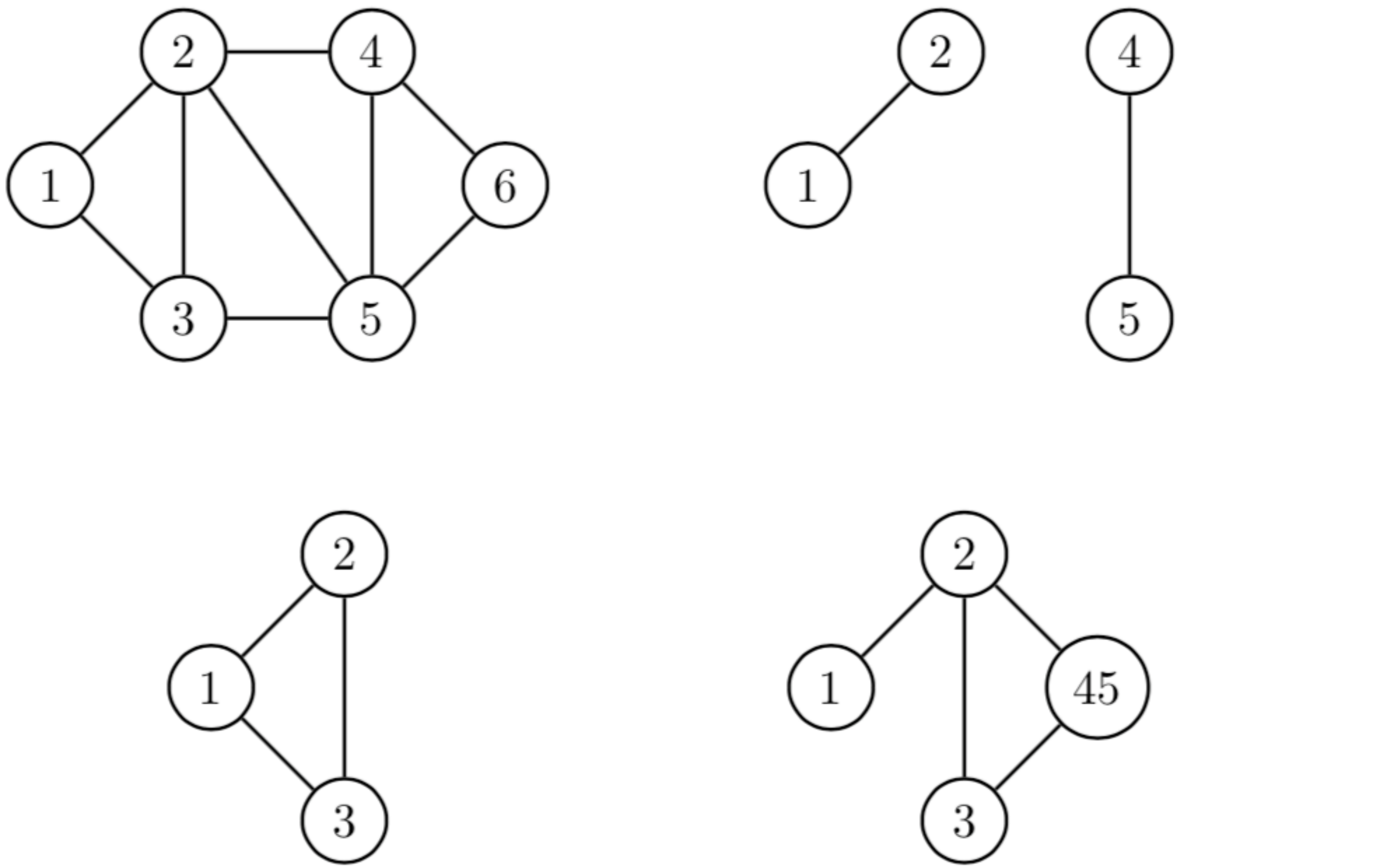


Figure 10.11: Examples of subgraphs and minors. Top left is the original graph. Top right is a subgraph that is not an induced subgraph. Bottom left is an induced subgraph. Bottom right is a minor.

Graph products

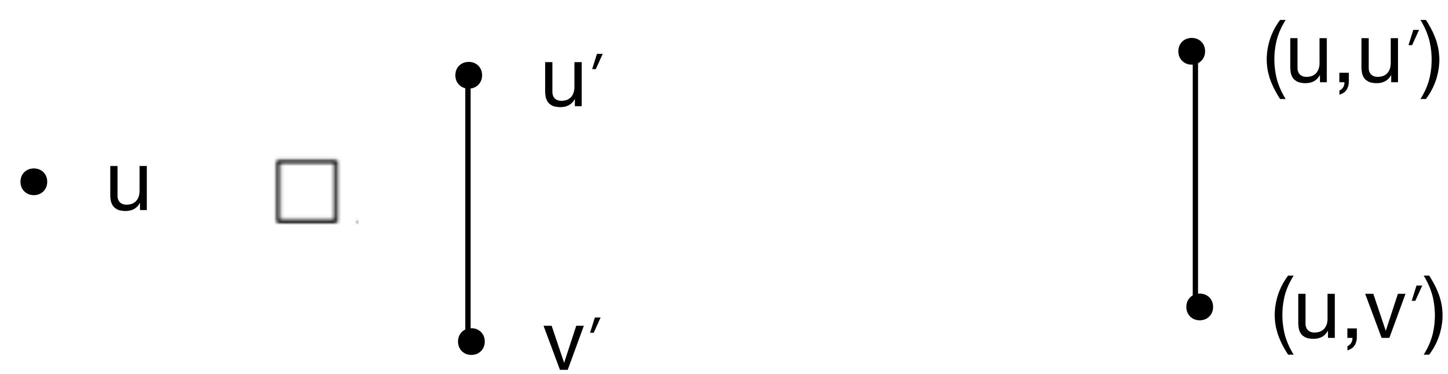
- There are at least five different definitions of the product of two graphs used by serious graph theorists. In each case the vertex set of the product is the Cartesian product of the vertex sets, but the different definitions throw in different sets of edges.

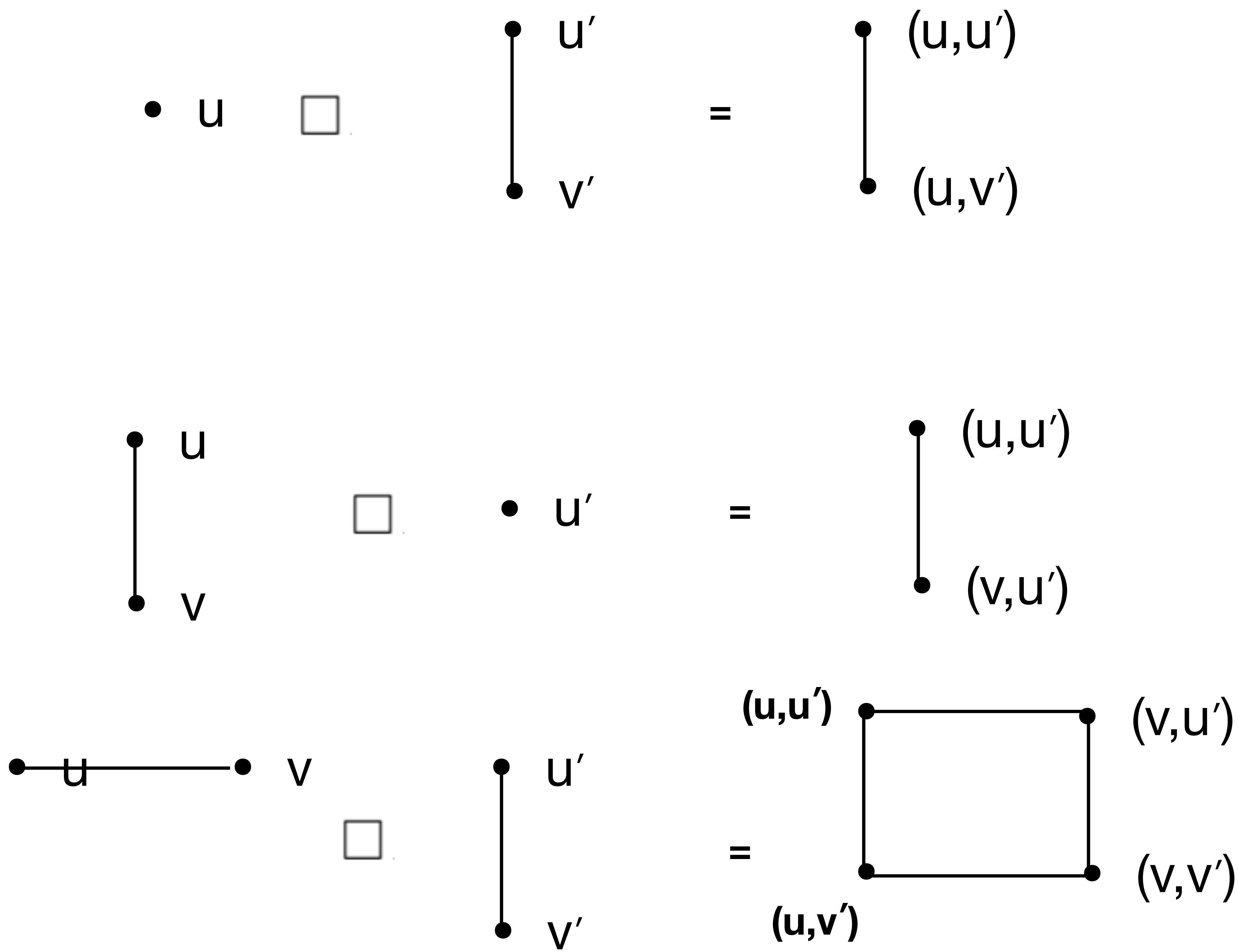
square product $\square$

- The square product or graph Cartesian product $G \square H$

- An edge (u,u′)(v,v′) is in $G \square H$ if and only if (a) u=v and u′v′ is an edge in H, or (b) uv is an edge in G and v = v′.

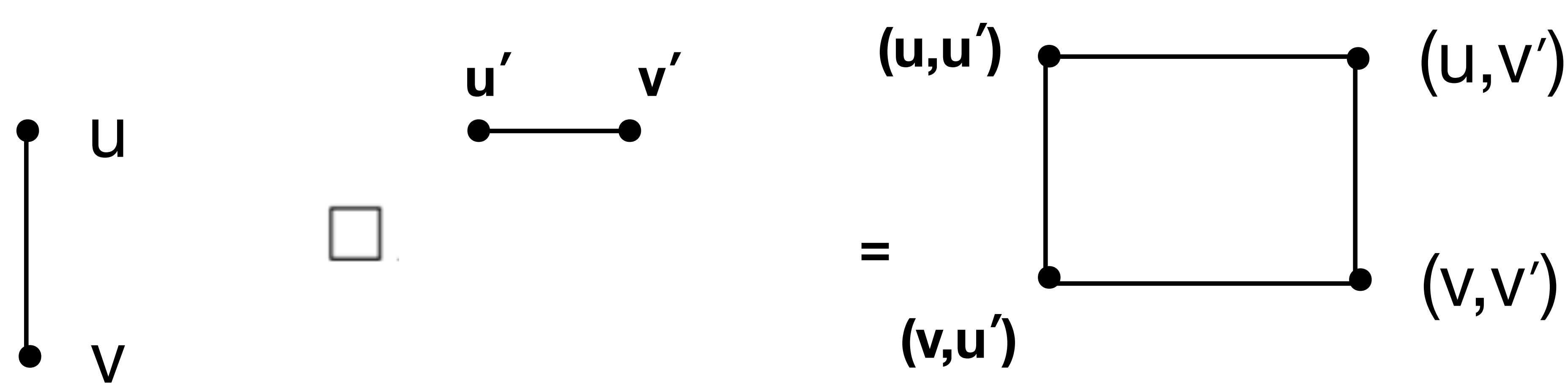
- It’s called the square product because the product of two (undirected) edges looks like a square.




u
u′
v′
=
(u,u′)
(u,v′)
u
v
u′
=
(u,u′)
(v,u′)
u
v
u′
v′
=
(u,u′)
(v,u′)
(u,v′)
(v,v′)

- The intuition is that each vertex in G is replaced by a copy of H, and then corresponding vertices in the different copies of H are linked whenever the original vertices in G are adjacent.

-



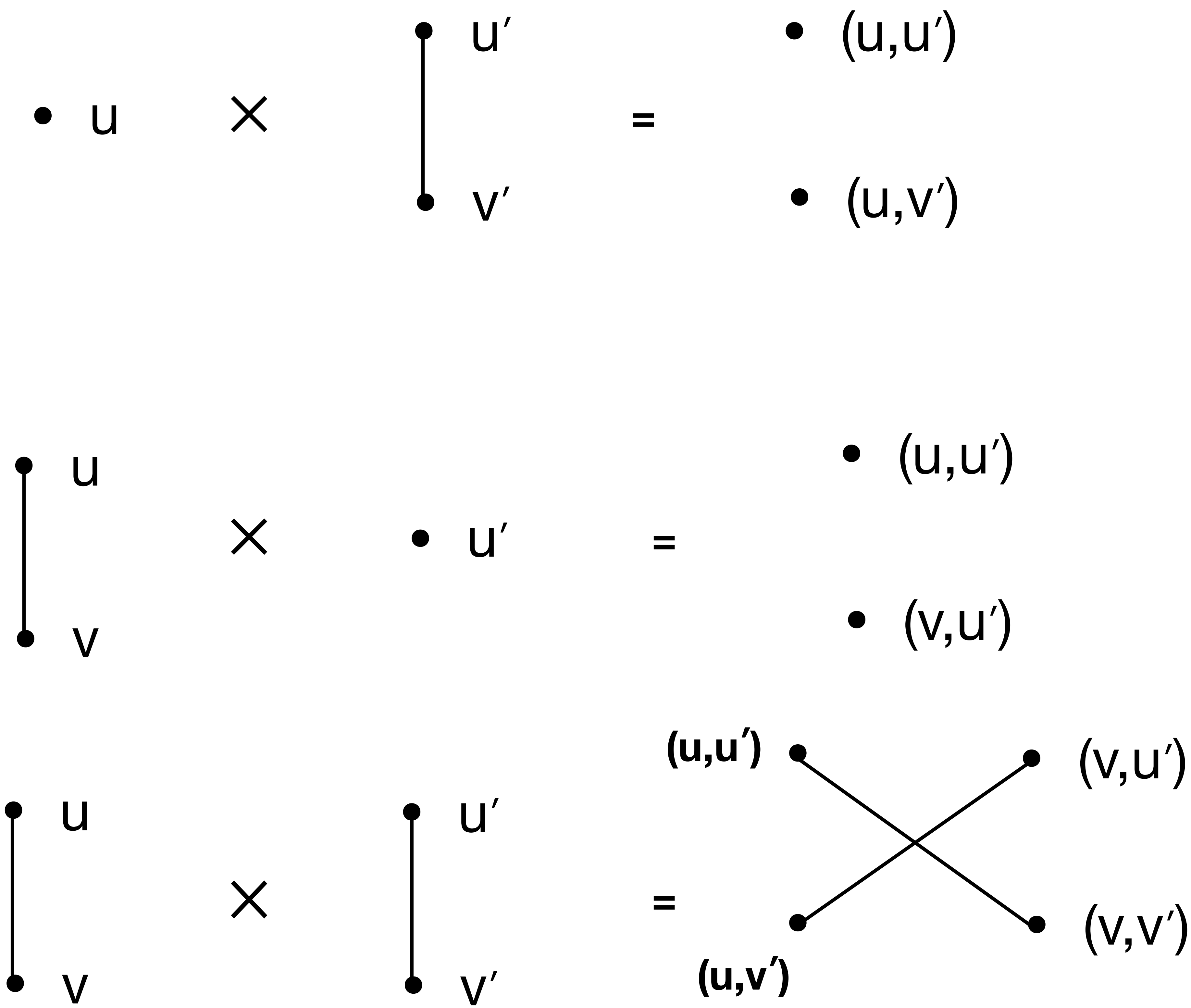
-

- The cube Q_n can be defined recursively by $Q_1 = P_1$ and $Q_n = Q_{n-1} \square Q_1$. It is also the case that $Q_n = Q_k \square Q_{n-k}$.

- An n-by-m **mesh** is given by $P_{n-1} \square P_{m-1}$.

- The cross product or categorical graph product $G \times H$
  - Now $(u,u')(v,v')$ is in $G \times H$ if and only if uv is in G and $u'v'$ is in H.
  - In the cross product, the product of two (again undirected) edges is a cross: an edge from (u, u') to (v, v') and one from (u, v') to (v, u').

Now $(u,u')(v,v')$ is in $G \times H$ if and only if uv is in G and $u'v'$ is in H.



- if C_1 and C_2 are the Cayley graphs of G_1 and G_2 (for particular choices of generators), then $C_1 \square C_2$ is the Cayley graph of $G_1 \times G_2$.

Functions between graphs

- A function from a graph G to another graph H typically maps V_G to V_H, with the edges coming along for the ride. For simplicity, we will generally write $f : G \rightarrow H$ when we really mean $f : V_G \rightarrow V_H$

同態

- A function f : G → H is a **graph homomorphism** if, for every edge uv in G, f(u)f(v) is an edge in H.

- Note that this only goes one way: it is possible to have an edge f(u)f(v) in H but no edge uv in G.

- Generally we will only be interested in functions between graphs that are homomorphisms, and even among homomorphisms, some functions are more interesting than others.

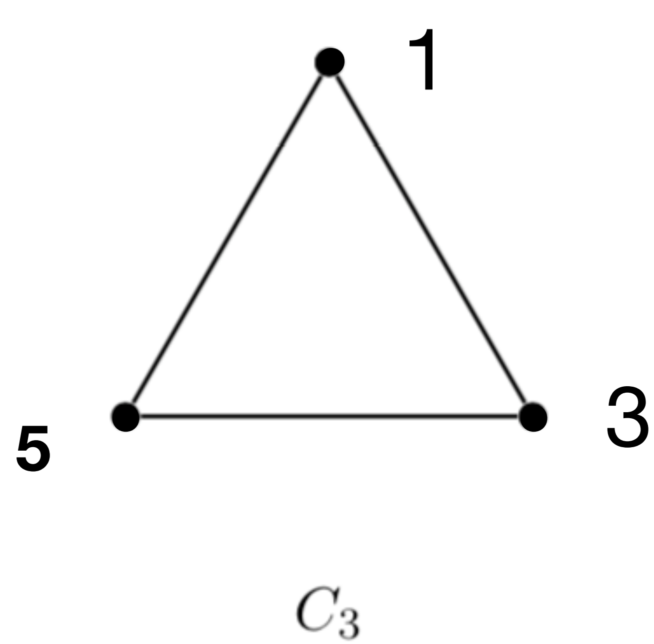
- A graph homomorphism that has an inverse that is also a graph homomorphism is called an **graph isomorphism**.

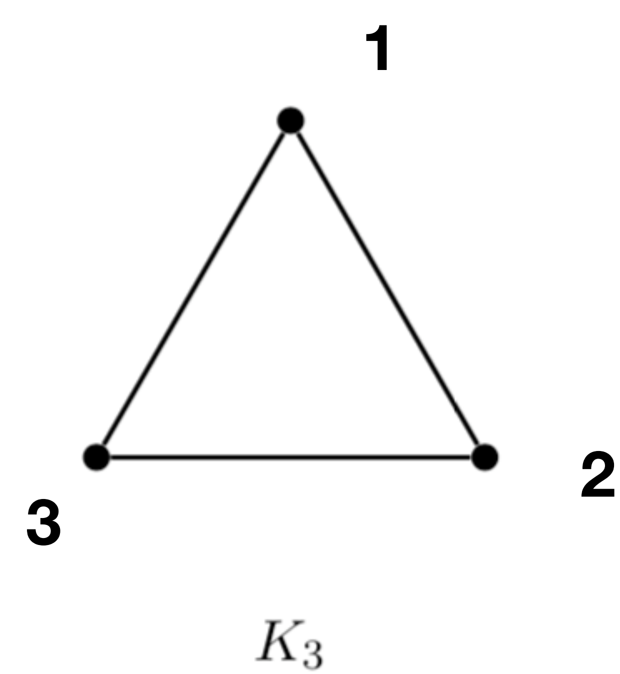
同構

- Two graphs G and H are **isomorphic** if there is an isomorphism between them.

- Intuitively, this means that G and H are basically the same graph, with different names for the vertices, and we will often treat them as the same graph.

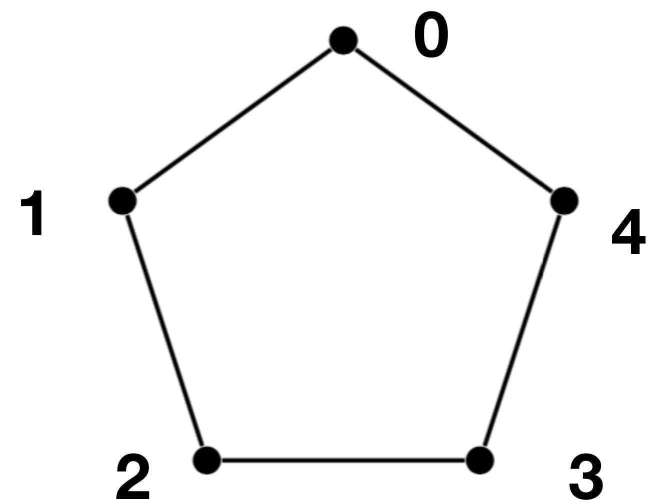
- for example, we will think of a graph G = (V,E) where V = {1,3,5} and E = {{1,3},{3,5},{1,5}} as an instance of C_3 and K_3 even if the vertex labels are not what we might have chosen by default. To avoid confusion with set equality, we write G ~= H when G and H are isomorphic.



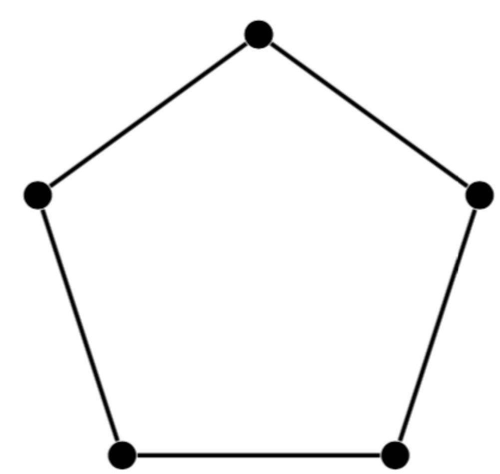


- Every graph is isomorphic to itself, because the identity function is an isomorphism. But some graphs have additional isomorphisms. An isomorphism from G to G is called an **automorphism** of G and corresponds to an internal symmetry of G. 自同構

- For example, the cycle C_n has 2n different automorphisms (to count them, observe there are n places we can send vertex 0 to, and having picked a place to send vertex 0 to, there are only 2 places to send vertex 1; so we have essentially n rotations times 2 for flipping or not flipping the graph). A path P_n (when $n > 1$) has 2 automorphisms (reverse the direction or not).



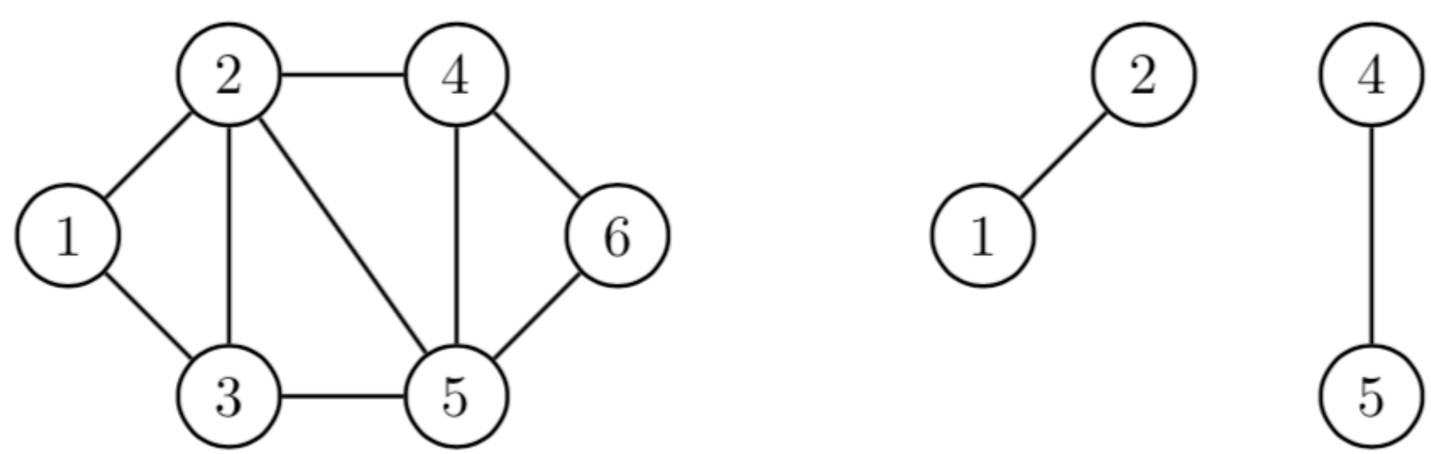
C_5



C_5

- An injective homomorphism from G to H is an isormophism between G and some subgraph H′ of H.

- In this case, we often say that G is a subgraph of H, even though technically it is just a copy of G that appears as a subgraph of H. This allows us to say, for example, that P_n is a subgraph of P_{n+1}, or all graphs on at most n vertices are subgraphs of K_n.

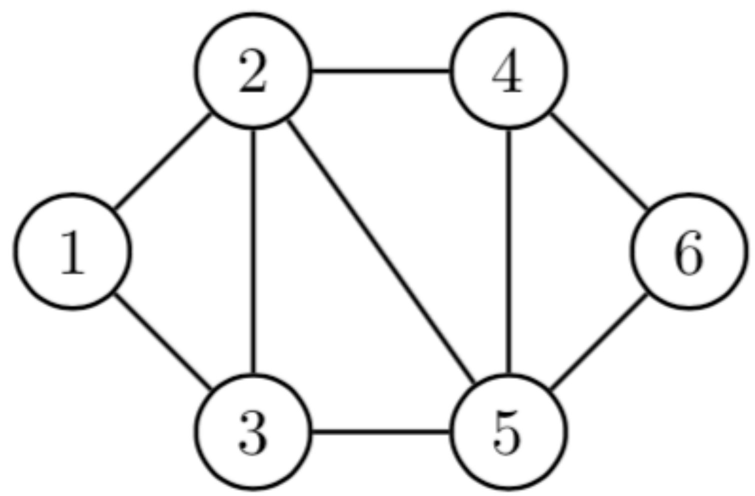


Paths and connectivity

- A fundamental property of graphs is connectivity: whether the graph can be divided into two or more pieces with no edges between them. Often it makes sense to talk about this in terms of reachability, or whether you can get from one vertex to another along some **path**.

- The pedantic definition of a path **path** of **length** n in a graph is the image of a homomorphism from P_n. In ordinary speech, it's a sequence of n+1 vertices $v_0,v_1,\ldots,v_n$ such that $v_i v_{i+1}$ is an edge in the graph for each i.

- A path is **simple** if the same vertex never appears twice (i.e. if the homomorphism is injective). If there is a path from u to v, there is a simple path from u to v obtained by removing cycles



a path : 4 2 5 3 2 1
a simple path: 4 2 1

- If there is a path from u to v, then v is **reachable** from u: u $\xrightarrow{*}$ v. We also say that u is **connected to** v.

- It's easy to see that connectivity is reflexive (take a path of length 0) and transitive (paste a path from u to v together with a path from v to w to get a path from u to w). But it's not necessarily symmetric if we have a directed graph.

- In an undirected graph, connectivity *is* symmetric, so it's an equivalence relation. The equivalence classes of $\xrightarrow{*}$ are called the **connected components** of G, and G itself is **connected** if and only if it has a single connected component, i.e., if every vertex is reachable from every other vertex.

- (Note that isolated vertices count as (separate) connected components.)

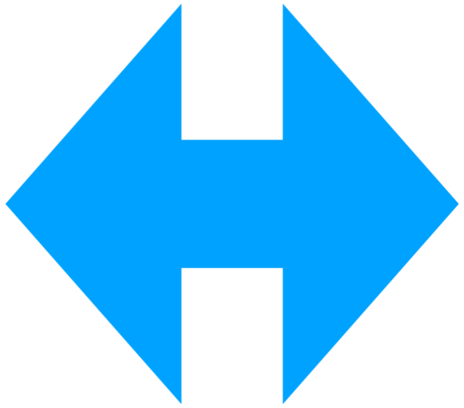
Theorem 9.4.1. *Let $\sim$ be a relation on A. Then each of the following conditions implies the others:*

1. *$\sim$ is reflexive, symmetric, and transitive.*
2. *There is a partition of A into disjoint* **equivalence classes** *A_i such that $x \sim y$ if and only if $x \in A_i$ and $y \in A_i$ for some i.*
3. *There is a set B and a function $f : A \to B$ such that $x \sim y$ if and only if $f(x) = f(y)$.*

- In a directed graph, we can make connectivity symmetric in one of two different ways:

- Define u to be **strongly connected** to v if u $\xrightarrow{*}$ v and v $\xrightarrow{*}$ u. I.e.,u and v are strongly connected if you can go from u to v and back again (not necessarily through the same vertices).

- It’s easy to see that strong connectivity is an equivalence relation. The equivalence classes are called **strongly-connected components**. A graph G is **strongly connected** if it has one strongly-connected component, i.e., if every vertex is reachable from every other vertex.



- Define u to be **weakly connected** to v if $u \stackrel{*}{\to} v$ in the undirected graph obtained by ignoring edge orientation.

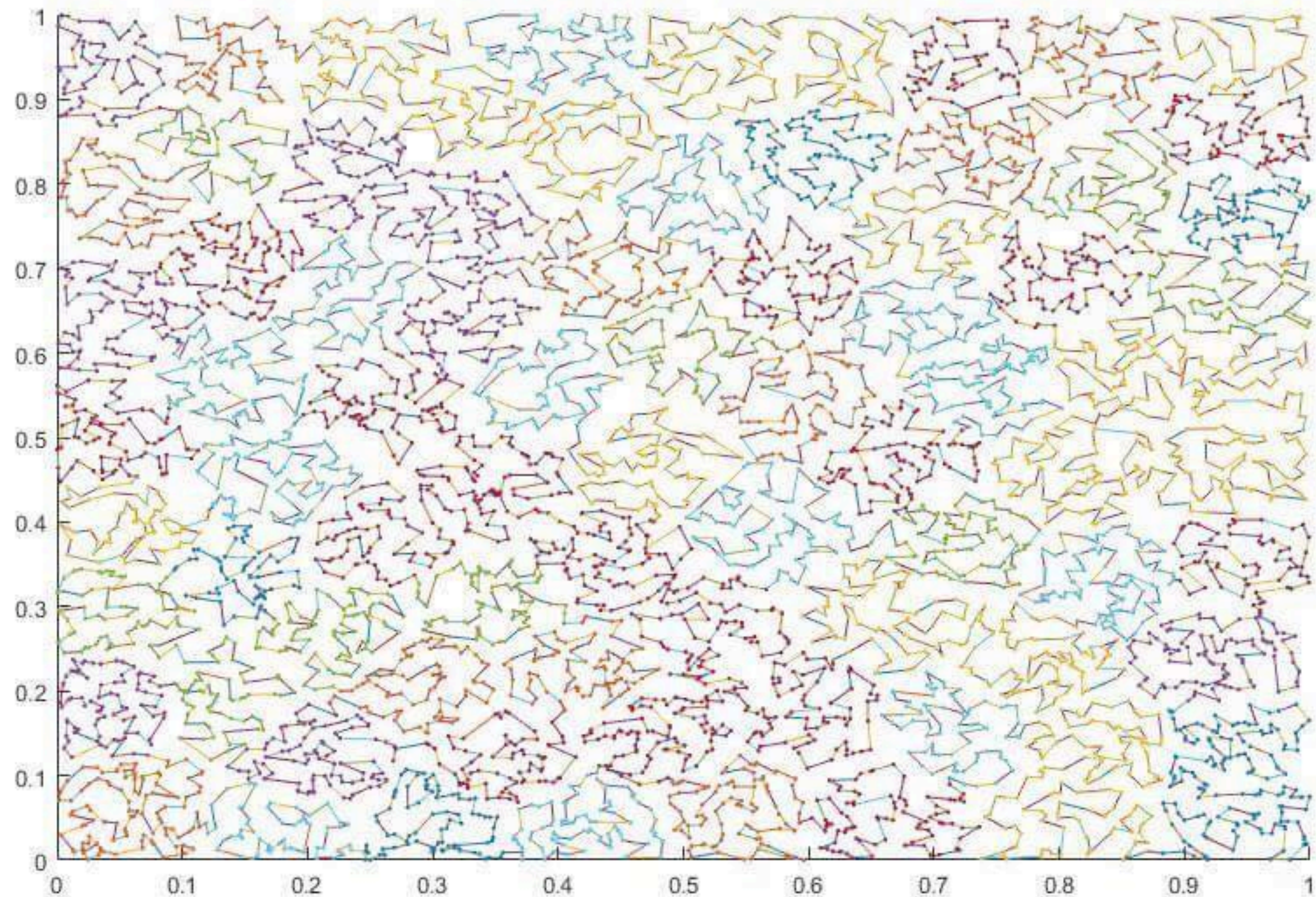
- The intuition is that u is weakly connected to v if there is a path from u to v if you are allowed to cross edges backwards.

- Weakly-connected components are defined by equivalence classes; a graph is weakly-connected if it has one component. Weak connectivity is a “weaker” property that strong connectivity in the sense that if u is strongly connected to v, then u is weakly connected to v; but the converse does not necessarily hold.

The k-th **power** G^k of a graph G has the same vertices as G, but uv is an edge in G^k if and only if there is a path of length k from u to v in G. The **transitive closure** of a directed graph: $G^* = \bigcup_{k=0}^{\infty} G^k$. I.e., there is an edge uv in G^* if and only if there is a path (of any length, including zero) from u to v in G, or in other words if $u \xrightarrow{*} v$. This is equivalent to taking the transitive closure of the adjacency relation.

Cycles

- The standard cycle graph C_n has vertices $\{0, 1, \ldots, n - 1\}$ with an edge from i to i+1 for each i and from n−1 to 0. To avoid degeneracies, n must be at least 3.
- A **simple cycle** of length n in a graph G is an embedding of C_n in G: this means a sequence of distinct vertices $v_0v_1v_2 \ldots v_{n-1}$, where each pair v_iv_{i+1} is an edge in G, as well as $v_{n-1}v_0$.
- If we omit the requirement that the vertices are distinct, but insist on distinct edges instead, we have a **cycle**. If we omit both requirements, we get a **closed walk**; this includes very non-cyclic-looking walks like the short excursion uvu. We will mostly worry about cycles.[2]



1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

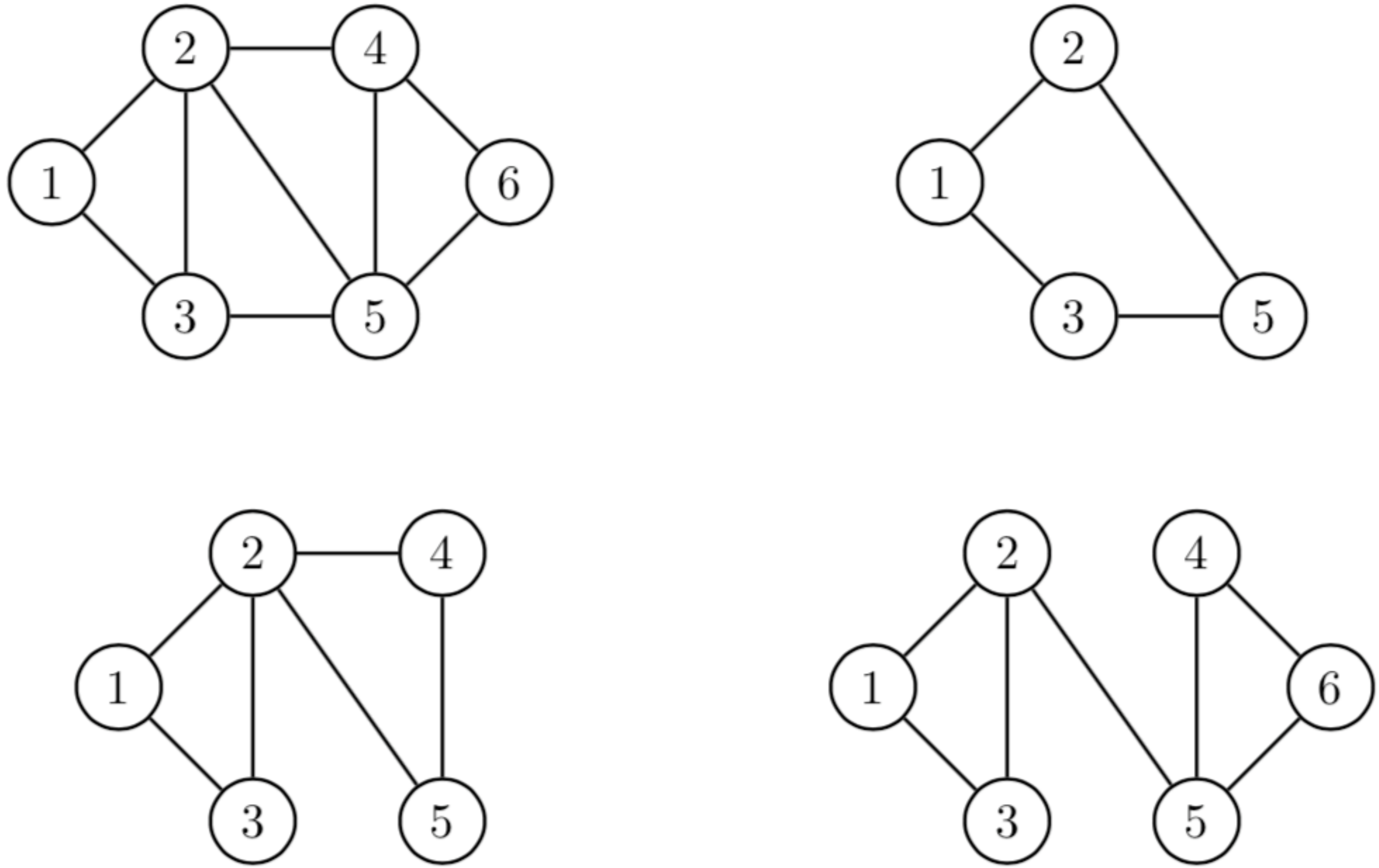


Figure 10.12: Examples of cycles and closed walks. Top left is a graph. Top right shows the simple cycle 1253 found in this graph. Bottom left shows the cycle 124523, which is not simple. Bottom right shows the closed walk 12546523, which uses the 25 edge twice.

- A graph with no cycles is **acyclic**. **Directed acyclic graphs** or **DAGs** have the property that their reachability relation $\stackrel{*}{\rightarrow}$ is a partial order; this is easily proven by showing that if $\stackrel{*}{\rightarrow}$ is not anti-symmetric, then there is a cycle consisting of the paths between two non-anti-symmetric vertices u $\stackrel{*}{\rightarrow}$ v and v $\stackrel{*}{\rightarrow}$ u.

- A **partial order** is a relation ≤ that is reflexive, transitive, and anti-symmetric. The last means that if x ≤ y and y ≤ x, then x = y.

- A relation R is **antisymmetric** if the only way that both (a,b) and (b,a) can be in R is if a=b.

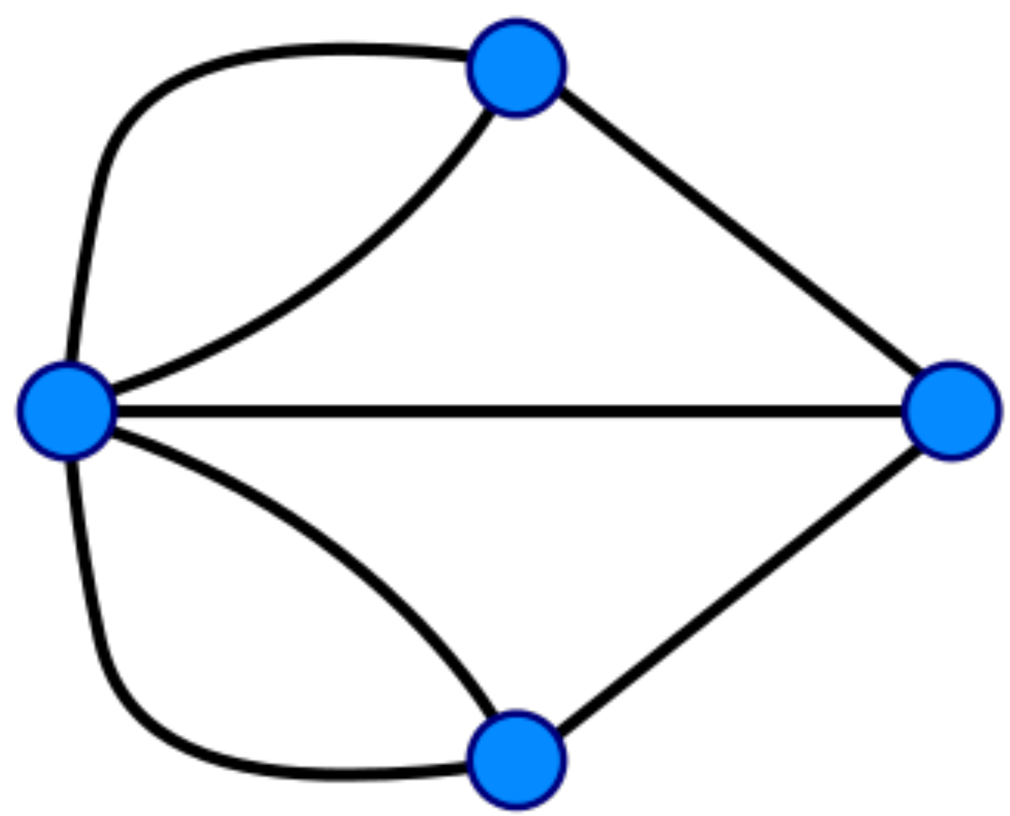
- Directed acyclic graphs may also be **topologically sorted**: their vertices ordered as $v_0, v_1, \ldots, v_{n-1}$, so that if there is an edge from v_i to v_j, then $i < j$.

- The proof is by induction on $|V|$, with the induction step setting v_{n-1} to equal some vertex with out-degree 0 and ordering the remaining vertices recursively. (See §9.5.5.1.)

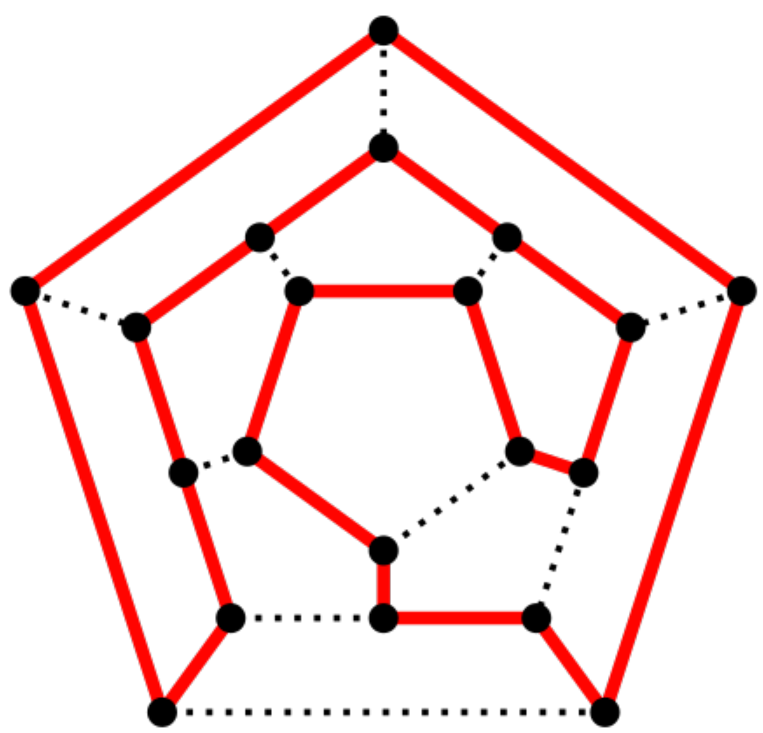
Trees

- Connected acyclic undirected graphs are called **trees.**
- A connected graph $G = (V, E)$ is a tree if and only if $|E| = |V| - 1$; we'll prove this and other characterizations of tree in §10.10.3.

- A cycle that includes every edge exactly once is called an **Eulerian cycle** or **Eulerian tour**, after Leonhard Euler, whose study of the *Seven bridges of Königsberg* problem led to the development of graph theory.



- A cycle that includes every vertex exactly once is called a **Hamiltonian cycle** or **Hamiltonian tour**, after William Rowan Hamilton, another historical graph- theory heavyweight (although he is more famous for inventing quaternions and the Hamiltonian).



- Graphs with Eulerian cycles have a simple characterization: a graph has an Eulerian cycle if and only if every vertex has even degree.

- Graphs with Hamiltonian cycles are harder to recognize.

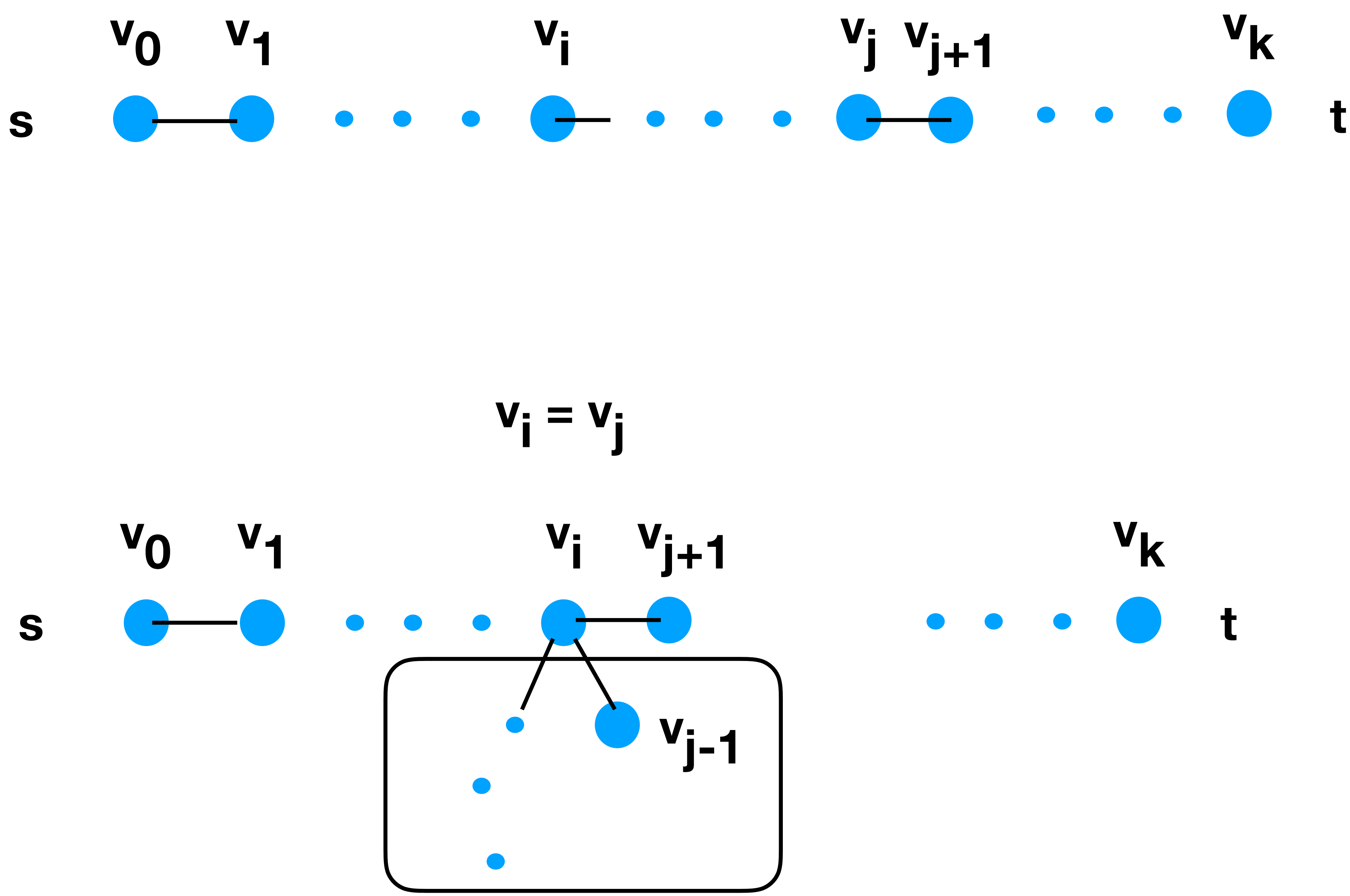
Paths and simple paths

- **Lemma 10.10.1.** If there is a path from s to t in G, there is a simple path from s to t in G.

- *Proof.* By induction on the length of the path. Specifically, we will show that if there is a path from s to t of length k, there is a simple path from s to t.

  - The base case is when $k = 1$; then the path consists of exactly one edge and is simple.

  - For larger k, let $s=v_0 \ldots v_k=t$ be a path in G. If this path is simple, we are done. Otherwise, there exist positions $i < j$ such that $v_i = v_j$. Construct a new path $v_1 \ldots v_i v_{j+1} \ldots v_k$; this is an s–t path of length less than k, so by the induction hypothesis a simple s–t path exists.



s
v_0
v_1
v_i
v_j
v_{j+1}
v_k
t
$v_i = v_j$
s
v_0
v_1
v_i
v_{j+1}
v_{j-1}
v_k
t

- **Lemma 10.10.2.** If there is a cycle in G, there is a simple cycle in G.

- *Proof.* As in the previous lemma, we prove that there exists a simple cycle if there is a cycle of length k for any k, by induction on k.

  - First observe that the smallest possible cycle has length 3, since anything shorter either doesn't get back to its starting point or violates the no-duplicate edges requirement.

- So the base case is k = 3, and it's easy to see that all 3-cycles are simple.

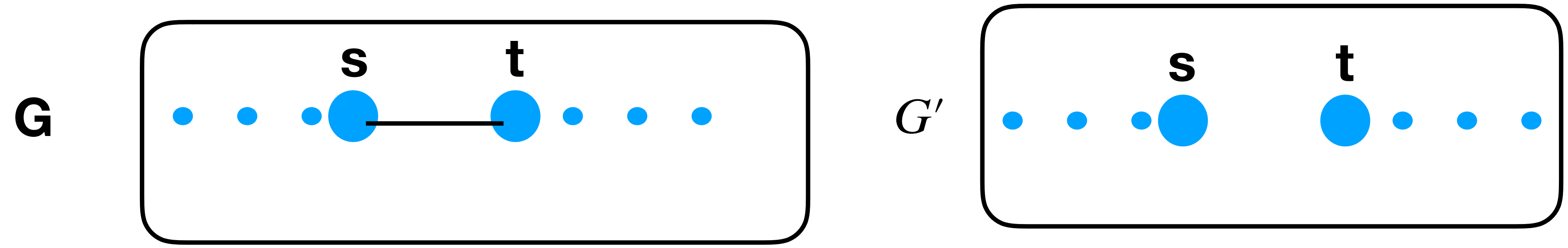
- For larger k, if $v_0 v_1 \dots v_{k-1}$ is a k-cycle that is not simple, there exist $i < j$ with $v_i = v_j$; patch the edges between them out to get a smaller cycle $v_0 \dots v_i v_{j+1} \dots v_{k-1}$. The induction hypothesis does the rest of the work.

The Handshaking Lemma

Lemma 10.10.3. *For any graph* $G = (V, E)$,

$$\sum_{v \in V} d(v) = 2|E|.$$

Proof. By induction on $m = |E|$. If $m = 0$, G has no edges, and $\sum_{v \in V} d(v) = \sum_{v \in V} 0 = 0 = 2m$. If $m > 0$, choose some edge st and let $G' = G - st$ be the subgraph of G obtained by removing st. Applying the induction hypothesis to G',



$$\begin{aligned}
2(m-1) &= \sum_{v\in V} d_{G'}(v) \\
&= \sum_{v\in V\setminus\{s,t\}} d_{G'}(v) + d_{G'}(s) + d_{G'}(t) \\
&= \sum_{v\in V\setminus\{s,t\}} d_G(v) + (d_G(s)-1) + (d_G(t)-1) \\
&= \sum_{v\in V} d_G(v) - 2.
\end{aligned}$$

So $\sum_{v\in V} d_G(v) - 2 = 2m - 2$, giving $\sum_{v\in V} d_G(v) = 2m$. □

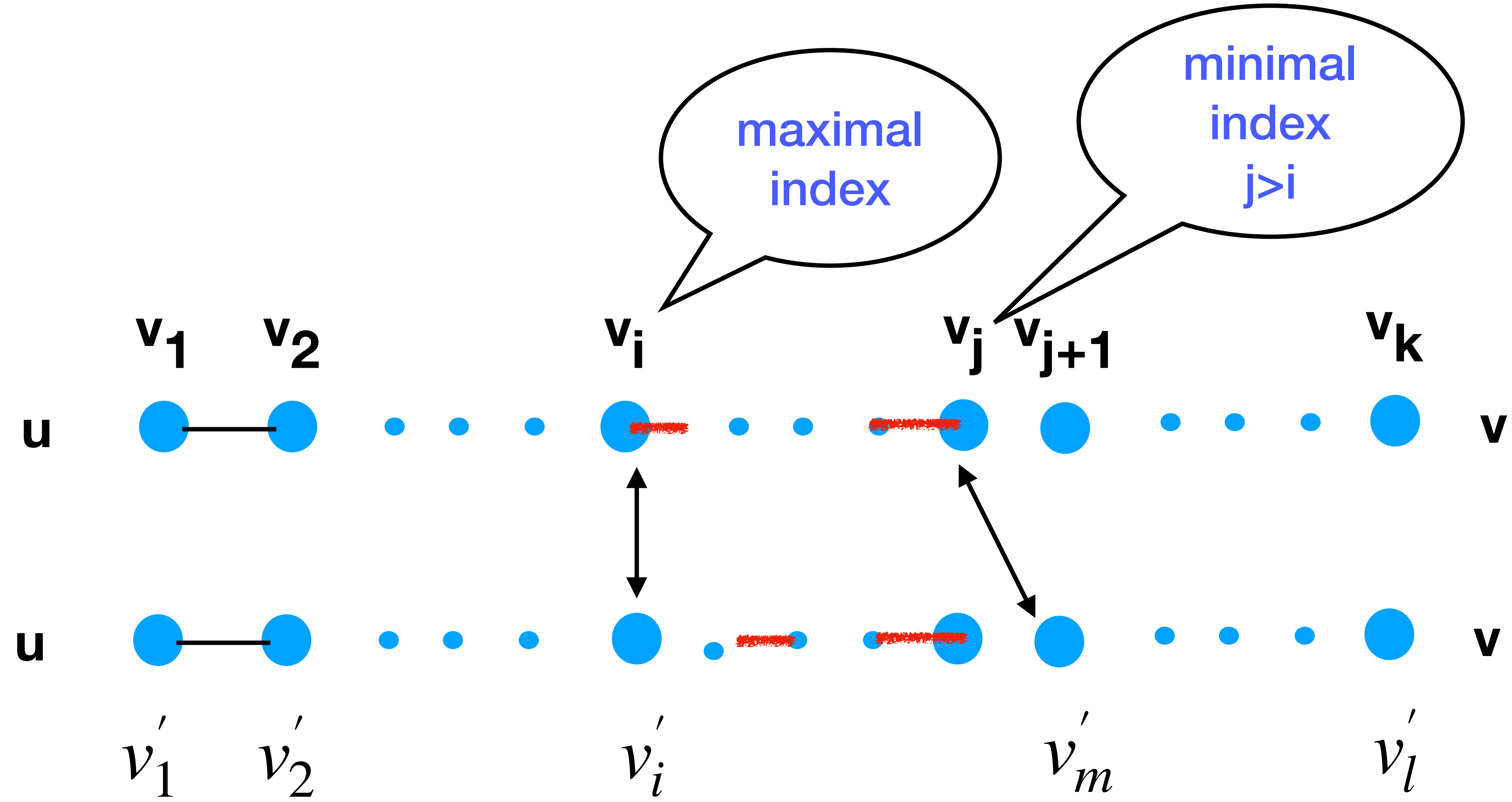
- A **tree** is defined to be an acyclic connected graph. There are several equivalent characterizations.

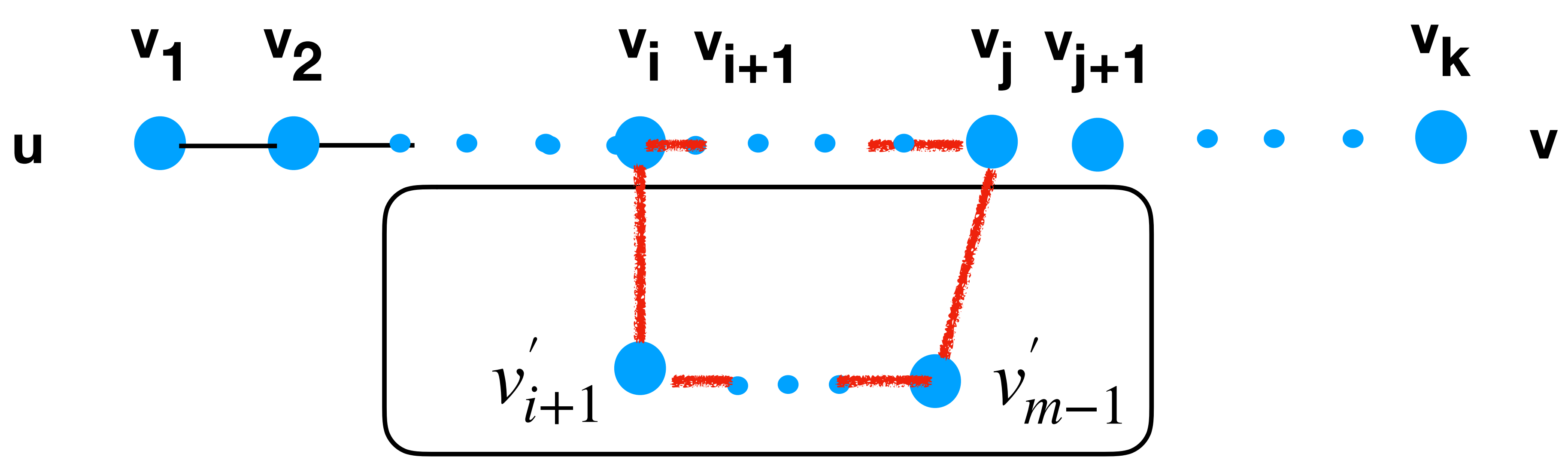
- **Theorem 10.10.4.** A graph is a tree if and only if there is exactly one simple path between any two distinct vertices.

- *Proof.* A graph G is connected if and only if there is *at least* one simple path between any two distinct vertices. We'll show that it is acyclic if and only if there is *at most* one simple path between any two distinct vertices.

- First, suppose that G has two distinct simple paths $u = v_1 v_2 \ldots v_k = v$ and $u = v_1' v_2' \ldots v_l' = v$. Let i be the largest index for which $v_i = v_i'$; under the assumption that the paths are distinct and simple, we have $i < \min(k, l)$.

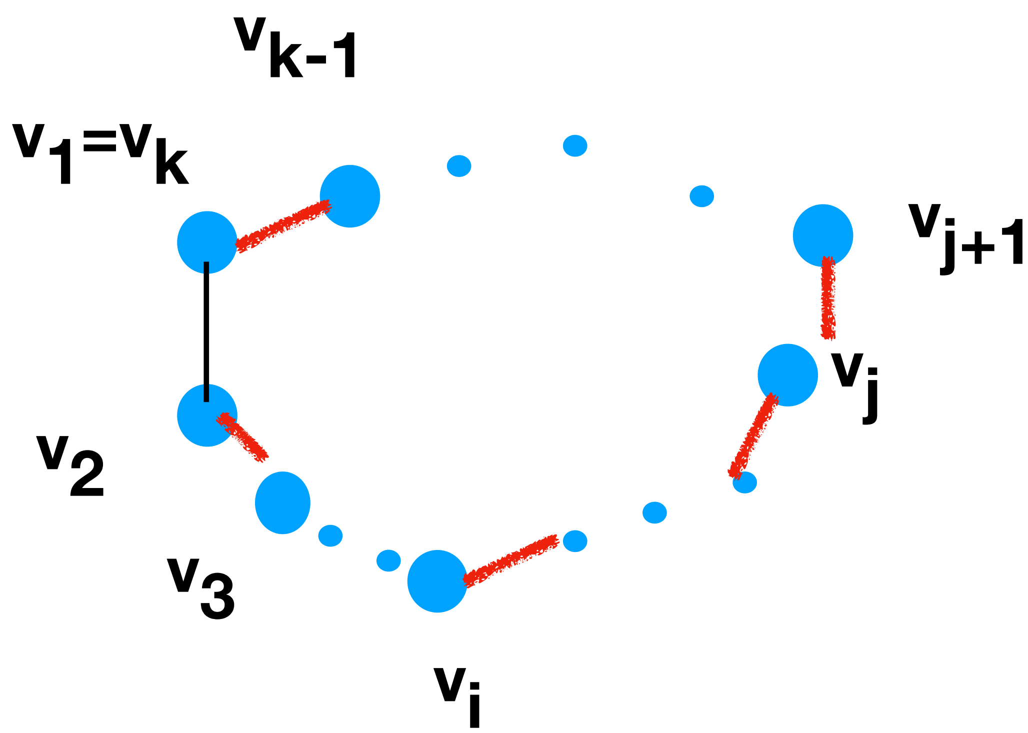
- Let $j > i$ be the smallest index for which $v_j = v_m'$ for some $m > i$; we know that some such j exists because, if nothing else, $v_k = v_l$. Let m be the smallest such m .





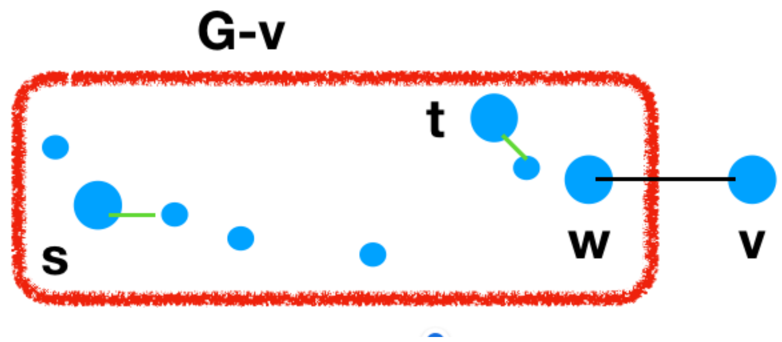
- a cycle $v_i v_{i+1} \ldots v_j V'_{m-1} V'_{m-2\ldots} V'_i = V_i$

- Now construct a cycle $v_i v_{i+1} \ldots v_j V'_{m-1} V'_{m-2 \ldots} V'_i = V_i$. This is in fact a simple cycle, since the v_r are all distinct, the v_s' are all distinct, and if any v_r with $i<r<j$ equals v_s' with $i<s<m$, then j or m is not minimal. It follows that if G has two distinct simple paths between the same vertices, it contains a simple cycle, and is not acyclic.



- Conversely, suppose that G is not acyclic, and let $v_1 v_2 \dots v_k = v_1$ be a simple cycle in G. Then $v_1 v_2$ and $v_2 \dots v_k$ are both simple paths between v_1 and v_2, one of which contains v_3 and one of which doesn't. So if G is not acyclic, it contains more than one simple path between some pair of vertices.

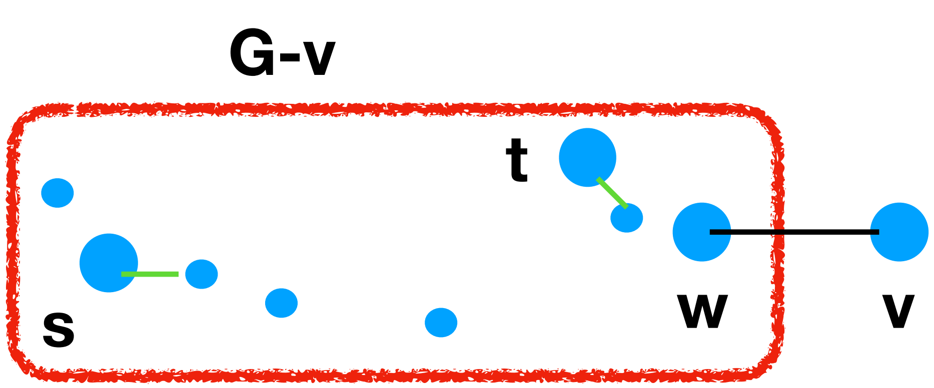
- An alternative characterization counts the number of edges: we will show
  - that any graph with less than $|V| - 1$ edges is disconnected, and
  - any graph with more than $|V| - 1$ edges is cyclic.
  - With exactly $|V| - 1$ edges, we will show that a graph is connected if and only if it is acyclic.



- **Lemma 10.10.5.** Let G be a nonempty graph, and let v be a vertex of G with d(v) = 1. Let G – v be the induced subgraph of G obtained by deleting v and its unique incident edge. Then
  - 1. G is connected if and only if G – v is connected.
  - 2. G is acyclic if and only if G – v is acyclic.

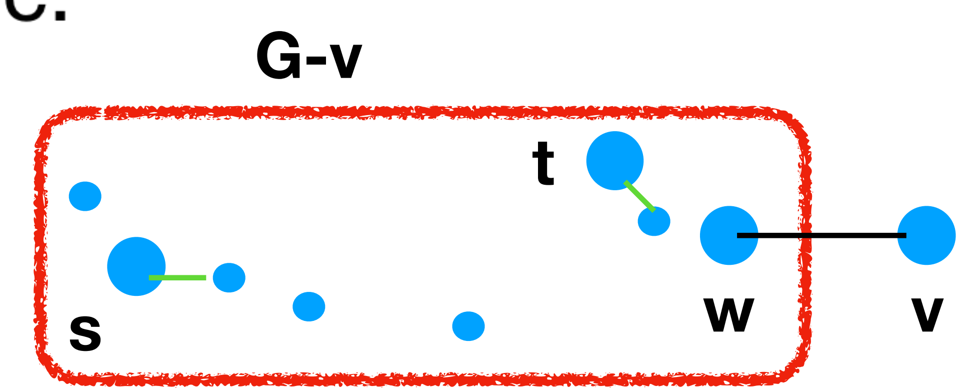
- 1. G is connected if and only if G – v is connected.

- *Proof.* Let w be v’s unique neighbor.
If G is connected, for any two vertices s and t, there is a simple s–t path. If neither s nor t is v, this path can’t include v, because w would appear both before and after v in the path, violating simplicity. So for any s, t in G–v, there is an s–t path in G–v, and G–v is connected.



A simple path between s and t must not include v since v only connects to w in G

2. G is acyclic if and only if G – v is acyclic.



- Conversely, if G – v is connected, then
    - any s and t not equal to v remain connected after adding vw, and
    - if s = v, for any t there is a path $w = v_1 \dots v_k = t$, from which we can construct a path $vv_1 \dots v_k = t$ from v to t.
    - The case t = v is symmetric.
- If G contains a cycle, then it contains a simple cycle; this cycle can't include v, so G – v also contains the cycle. Conversely, if G – v contains a cycle, this cycle is also in G.

Lemma 10.10.3. *For any graph $G = (V, E)$,*

$$\sum_{v \in V} d(v) = 2|E|.$$

- Corollary 10.10.6. *Let* G = (V, E)*. If* |E| < |V | – 1, G *is not connected.*

- *Proof.* By induction on n = |V |.
  For the base case, if n = 0, then |E| = 0 $\not<$ n – 1.
  For larger n, suppose that n ≥ 1 and |E| < n – 1. From Lemma 10.10.3 we have $\sum_v$ d(v) < 2n – 2, from which it follows that there must be at least one vertex v with d(v) < 2.

- If d(v) = 0, then G is not connected. If d(v) = 1, then G is connected if and only if G–v is connected. But G–v has n–1 vertices and |E| – 1 < n – 2 edges, so by the induction hypothesis, G – v is not connected.

- So in either case, |E| < n – 1 implies G is not connected.

- In the other direction, combining the lemma with the fact that the unique graph K_3 with three vertices and at least three edges is cyclic tells us that any graph with at least as many edges as vertices is cyclic.

- Corollary 10.10.7. *Let* G = (V, E)*. If* |E| > |V | − 1*,* G *contains a cycle.*

- *Proof.* By induction on n = |V |.
  For n ≤ 2, |E| ≯ |V − 1|, so the claim holds vacuously.[3] For larger n, there are two cases:

- 1. Some vertex v has degree d(v) ≤ 1. Let G′ = (V′, E′) = G − v. Then |E′| ≥ |E|−1 > |V|−2 = |V′|−1, and by the induction hypothesis G′ contains a cycle. This cycle is also in G.

- 2. Every vertex v in G has d(v) ≥ 2. Let's go for a walk: starting at some vertex v_0, choose at each step a vertex v_{i+1} adjacent to v_i that

- does not already appear in the walk. This process finishes when we reach a node v_k all of whose neighbors appear in the walk in a previous position. One of these neighbors may be v_{k-1}; but since $d(v_k) \geq 2$, there is another neighbor $v_j \neq v_{k-1}$. So $v_j \ldots v_k v_j$ forms a cycle.

- **Theorem 10.10.8.** Let $G = (V, E)$ be a nonempty graph. Then any two of the following statements implies the third:

- 1. G is connected.
  2. G is acyclic.
  3. $|E|=|V|-1$.

1. G is connected.
2. G is acyclic.
3. $|E|=|V|-1$.

- *Proof.* We will use induction on n for some parts of the proof. The base case is when $n = 1$; then all three statements hold always. For larger n, we show:

- (1) and (2) imply (3): Use Corollary 10.10.6 and Corollary 10.10.7.

- (1) and (3) imply (2). From Lemma 10.10.3, $\sum_{v \in V} d(v) = 2(n-1) < 2n$. It follows that there is at least one v with $d(v) \leq 1$. Because G is connected, we must have $d(v) = 1$. So $G' = G-v$ is a graph with $n-2$ edges and $n-1$ vertices. It is connected by Lemma 10.10.5, and thus it is acyclic by the induction hypothesis. Applying the other case of Lemma 10.10.5 in the other direction shows G is also acyclic.

Corollary 10.10.6. *Let* $G = (V, E)$*. If* $|E| < |V| - 1$, G *is not connected.*

Corollary 10.10.7. *Let* $G = (V, E)$*. If* $|E| > |V| - 1$, G *contains a cycle.*

1. G is connected.
2. G is acyclic.
3. $|E|=|V|-1$.

-

- (1) and (3) imply (2). From Lemma 10.10.3, $\sum_{v \in V} d(v) = 2(n-1) < 2n$. It follows that there is at least one v with $d(v) \leq 1$. Because G is connected, we must have $d(v) = 1$. So $G' = G-v$ is a graph with $n-2$ edges and $n-1$ vertices. It is connected by Lemma 10.10.5, and thus it is acyclic by the induction hypothesis. Applying the other case of Lemma 10.10.5 in the other direction shows G is also acyclic.

- **Lemma 10.10.5.** Let G be a nonempty graph, and let v be a vertex of G with $d(v) = 1$. Let $G - v$ be the induced subgraph of G obtained by deleting v and its unique incident edge. Then
  - 1. G is connected if and only if $G - v$ is connected.
  - 2. G is acyclic if and only if $G - v$ is acyclic.

Corollary 10.10.7. *Let* G = (V, E)*. If* |E| > |V | − 1*,* G *contains a cycle.*

1. G is connected.
2. G is acyclic.
3. |E|=|V|−1.

- (2) and (3) imply (1). As in the previous case, G contains a vertex v with d(v) ≤ 1. If d(v) = 1, then G−v is a nonempty graph with n − 2 edges and n − 1 vertices that is acyclic by Lemma 10.10.5. It is thus connected by the induction hypothesis, so G is also connected by Lemma10.10.5. If d(v)=0, then G−v has n−1 edges and n−1 vertices. From Corollary 10.10.7, G − v contains a cycle, contradicting (2).

G-v satisfies conditions (2) and (3). By induction hypothesis, G-v is connected

- **Lemma 10.10.5.** Let G be a nonempty graph, and let v be a vertex of G with d(v) = 1. Let G − v be the induced subgraph of G obtained by deleting v and its unique incident edge. Then
  - 1. G is connected if and only if G − v is connected.
  - 2. G is acyclic if and only if G − v is acyclic.